TRANSPORTATION ANALYSIS SIMULATION SYSTEM (TRANSIMS)

VERSION 1.0

Microsimulation

P. Stretz
Military Systems Analysis and Simulations Group
Los Alamos National Laboratory

K. P. Berkbigler
Computer Research and Applications Group
Los Alamos National Laboratory

March 1998

[Tab 4]

Contents

1. INTRODUCTION

The TRANSIMS Microsimulation component simulates the movement and interactions of travelers in the transportation system of a metropolitan region. Using a trip plan provided by the route planner, each traveler attempts to execute the plan on the transportation system. The combined traveler interactions produce emergent behaviors such as traffic congestion.

The microsimulation used for Interim Operational Capability 1 (IOC-1) simulates vehicular travelers in which each vehicle contains one traveler. Intermodal travel plans and multiple travelers per vehicle will be represented in future TRANSIMS IOCs. Roadway transportation was chosen for the initial emphasis because of its high use, complexity, and importance to air quality. The roadway network includes freeways, highways, streets, ramps, turn pocket lanes, and intersections (signalized and unsignalized). Vehicles executing trip plans accelerate, decelerate, turn, change lanes, pass, and respond to other vehicles, signs, and signals.

The microsimulation uses a cellular automata approach to provide the computational speed necessary to simulate an entire region at the individual traveler level. The cellular automata technique provides a means to simulate large numbers of vehicles and maintain a fast execution speed. Each link in the transportation network is divided into a finite number of cells. At each timestep of the simulation, each cell is examined for a vehicle occupant. If a vehicle is present in the cell, the vehicle may be advanced to another cell using a simple rule set. Increasing the fidelity by decreasing the cell size, adding vehicle attributes, and expanding the rule set results in slower computational speed. The fidelity and performance limits of the cellular automata microsimulation are evaluated to establish the computational detail required to support the fidelity necessary to meet analysis requirements.

2. TRANSPORTATION NETWORK

2.1 Simulation Study and Buffer Links

The microsimulation distinguishes two types of links in its calculations:

1) study area links, and

2) buffer area links.

Study area links are the links of interest for the traffic analyst. The output subsystem, for example, records events such as when a vehicle leaves or enters the study area. The nature of the microsimulation makes it necessary to simulate traffic on additional buffer area links. Typically, these links form a *fringe* about two links in width around the study area. A simulation includes buffer links in order to avoid *edge effects* such as when vehicles enter the study area on its boundary. Buffer area links give these vehicles time to interact with other traffic and achieve realistic behavior before entering the study area. The analyst must choose appropriate buffer area links for a given study area.

2.2 Traffic Controls

This version of TRANSIMS implements pre-timed signals only. Signalized traffic controls are initialized at the beginning of the simulation to the first interval of the first phase of the signal cycle when the signal offset is 0.0. When the offset is non-zero, the signal is initialized to the phase and interval that corresponds to simulation time 0 in the offset cycle. Signals are updated at each timestep according to the timing table provided for each signal.

2.3 Pocket Lanes

A pocket lane is a lane that does not extend the whole length of a link. A turn pocket starts mid-link and ends at the intersection, whereas a merge pocket starts at the intersection and ends mid-link. A link may have a left or right turn pocket to facilitate turns at an intersection. Merge pockets facilitate entry onto a link.

In the microsimulation, pocket lanes are represented similarly to through lanes, except that cells in a pocket lane that are not part of a pocket are marked as unusable for traffic in the grid.

3. TRAFFIC DYNAMICS

Traffic dynamics in the microsimulation are produced by interactions of individual vehicles on the transportation network. The position of vehicles on the roadway is determined by applying a rule set that governs movement and lane changes. This rule set must be as simple as possible in order to maintain the computational speed necessary to update positions of the large number of vehicles that could be present in a regional traffic microsimulation. The rule set imposes a no-collision strategy on the vehicles. Vehicle interactions based on the rule set combine to produce emergent driver behavior. Traffic dynamics require that for any vehicle *v* at time *t*, all position change calculations must be based on other vehicle positions at time *t*, not at time *t + 1*.

3.1 Definitions

Timestep One microsimulation update cycle in which all movement and lane changes are executed for each vehicle. Each Timestep typically represents approximately one second of simulation time.

Grid Division of the link into cells forming a grid (Figure 1). The microsimulation uses a separate grid for each lane on the roadway. Each cell is 7.5 meters long.

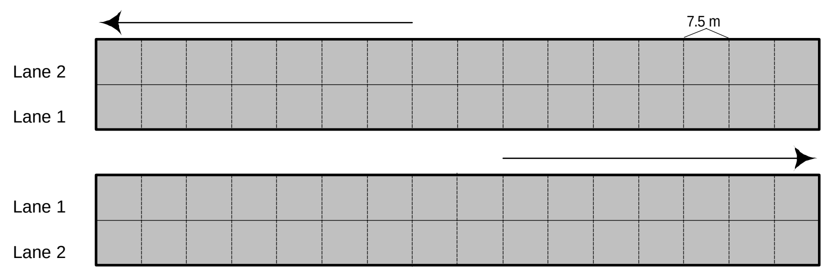

Figure 1: Link with Grid Cells

Gap Number of empty cells between this vehicle and the next vehicle on the grid (Figure 2). If this is the first vehicle on the grid, gap is the number of empty cells between this vehicle and the end of the grid.

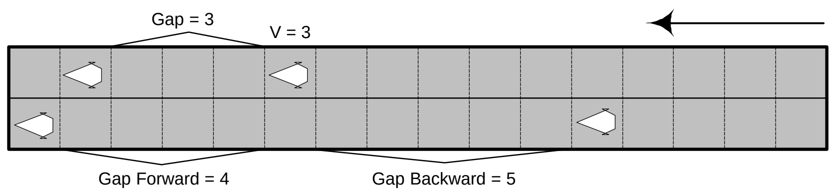

Figure 2: Gaps

V Speed of the vehicle in cells/timestep.

V_{Max} Speed limit on the link in cells/timestep.

$V_{GlobalMax}$	Maximum speed on any link in cells/timestep. Set to five cells/timestep.
P_D	Deceleration probability. Probability that a vehicle will decelerate during a timestep.
D_{PF}	Distance from the intersection where a vehicle starts to consider changing lanes in order to follow its plan. Default value is 525 meters (70 grid cells).
N_{rand}	Random number between 0.0 and 1.0.

3.2 Movement

The movement rule is "Accelerate when you can; slow down if you must; sometimes slow down for no reason." The rule is executed to update the speed and position of each vehicle on the roadway.

The distance between a vehicle and the next car ahead is called the *gap*. Each vehicle will try to accelerate if the gap is greater than the desired speed. The desired speed is limited to the speed limit posted on each link. If the gap is smaller than the current speed, the vehicle will slow down until its current speed is equal to the gap, thus imposing the no-collision condition. Each vehicle also has a random probability of slowing down. This is called the deceleration probability (P_D). Use of the deceleration probability is essential to produce realistic traffic dynamics, such as jam waves, from the individual vehicle interactions.

To compute a vehicle's speed (V_{t+1}) and the next position on a link, first compute the speed based on the gap and the vehicle's speed in the current timestep (V_t).

1) Compute Gap

2) if (V_t < Gap AND $V_t < V_{Max}$)
$$V_{t+1} = V_t + 1$$

Each moving vehicle (Speed > 0) has a random probability of decelerating in each timestep. Compute the probability and slow down if the computed probability is less than the deceleration probability.

3) if ($V_{t+1} > 0$) and ($N_{Rand} < P_D$)
$$V_{t+1} = V_{t+1} - 1$$

Next, move the vehicle to its new grid position based on the new speed.

$$\text{New Cell} = \text{Current Cell} + V_{t+1}$$

3.3 Lane Changes

Vehicles will change lanes for two reasons:

1) to pass a slower vehicle in the current lane

2) to make turns at intersections in order to follow its plan

The decision to make a lane change in order to pass a slower vehicle is based on the gap on the current lane, the gap backward on the new lane, and the gap forward on the new lane.

A vehicle that needs to make a turn at the next intersection in order to follow its plan will start to consider a lane change when it is within a set distance from the intersection. As the vehicle approaches the intersection, the urgency to change into a lane that is appropriate for plan following increases as the vehicle approaches the intersection. Any vehicles that fail to make the required lane changes for plan following are marked and removed at the nearest parking place.

Lane changes are made before movement calculations in the microsimulation update cycle. Left and right lane changes are made on alternating timesteps to ensure that gap calculations are based on vehicle positions at time *t*, not *t+1*. Left lane changes are made on even timesteps; right lane changes are made on odd timesteps. Multi-lane roadways are processed from left to right when making left lane changes and from right to left when making right lane changes.

3.3.1 Passing Lane Change

Passing lane changes are based on three gap calculations (Figure 2):

1) Gap in the current lane (G_c)

2) Gap forward in the new lane (G_f)

3) Gap backward in the new lane (G_b)

These gaps are used to set the weight values (Table 1) that are in the calculations to determine if a lane change will be made. The calculated gaps and the potential speed of the vehicle in the current timestep are used to determine whether a vehicle will make a lane change for passing.

Table 1: Weight Values

Parameter	Description	Equation
Weight 1	An integer value based on the gap in the current lane, the potential speed of the vehicle in this timestep, and the gap forward in the new lane.	Weight 1 = $((V+1 > G_c)$ AND $(G_f > G_c))$
Weight 2	An integer value based on the gap forward in the new lane and the speed of the vehicle.	Weight 2 = $V - G_f$
Weight 3	An integer value based on the gap backward in the new lane and the maximum speed of a vehicle in the simulation.	Weight 3 = $V_{GlobalMax} - G_b$
Weight 4	An integer value based on the distance from the intersection (D_I) and point on the link where a vehicle starts to consider lane changes to follow its plan (D_{PF}).	Weight 4 = $(D_{PF} - D_I)/5$

A vehicle will make a lane change for passing if the following three conditions are satisfied:

1) Weight 1 > 0

2) Weight 1 > Weight 2

3) Weight 1 > Weight 3

3.3.2 Plan Following Lane Change

Acceptable approach lanes that allow a vehicle to transition to the next link in its plan are determined when a vehicle enters a link. Lane changes for plan following are introduced into the lane change calculations when a vehicle is within a set distance from an intersection (D_{PF}). The bias to make a plan following lane change increases as the vehicle nears the intersection. If the vehicle is already in an acceptable approach lane, the vehicle is biased to stay in the correct lane and ignore lane changes to pass slower vehicles (i.e., lane changes based on gaps).

Plan following lane changes are controlled by introduction of an additional parameter to the lane change calculations. The parameter, Weight 4, is initially set to zero.

If the vehicle is within the D_{PF} and is not in an acceptable approach lane, Weight 4 is set based on the distance between the vehicle and the intersection (D_I). Weight 4 increases as the vehicle moves nearer to the intersection.

Since only one type of lane change is made during a timestep, the type of lane change needed (left/right) must be the same as the type of lane change (left/right) that is calculated during this timestep.

If a vehicle is already in an acceptable approach lane and is within the D_{PF}, Weight 4 = -1, which will prevent any lane changes based on gaps (passing lane changes).

It is possible for a vehicle to have more than one approach lane that is acceptable for plan following. If the vehicle is in an acceptable lane and the new lane (left/right) is also an acceptable approach lane, Weight 4 = 0, which allows lane changes based on gaps.

The final calculation to determine the lane change calculates Weight 1 using Weight 4.

$$\text{Weight 1} = ((V+1 > G_c) \text{ AND } (G_f > G_c)) + \text{Weight 4}$$

Weights 2 and 3 are calculated as indicated in the passing lane changes above.

A vehicle will make a lane change if the following three conditions are satisfied:

1) Weight 1 > 0

2) Weight 1 > Weight 2

3) Weight 1 > Weight 3

3.3.3 Special Cases – Merge Lanes, Turn Pocket Lanes, Look-Ahead Across Links

3.3.3.1 Merge Lanes

Vehicles in merge lanes are forced to make lane changes in the same direction as the merge direction. In some cases, a lane can have a merge pocket and a turn pocket further down the lane toward the intersection. In this case, vehicles are prohibited from entering the lane until they are past the end point of the merge pocket.

3.3.3.2 Turn Pocket Lanes

Speed restrictions are imposed on a vehicle attempting to enter a turn pocket lane from the adjacent lane. These restrictions prevent movement of the vehicle past the start of the turn pocket, which causes the vehicles to queue on the adjacent lane until a lane change into the turn pocket lane is possible.

In Figure 3, the vehicle in Lane 2 needs to make a left turn at the next intersection. The left turn pocket (Lane 1) has no vacant cells. At time t, the vehicle’s speed is 3, which will move the vehicle past the start of the turn pocket. The vehicle’s speed is constrained to 2 (the distance from the vehicle’s current position at time t and the start of the turn pocket). At time t+1, the vehicle has moved down Lane 2 to the starting cell of the turn pocket. A lane change into the turn pocket is not possible because all of the cells are occupied by other vehicles. The vehicle is prevented from traveling further down Lane 2 by constraining the speed to 0. At time t+2, the vehicle remains on Lane 2 with speed 0. The vehicle’s speed will remain constrained to 0 until a lane change into the turn pocket is possible.

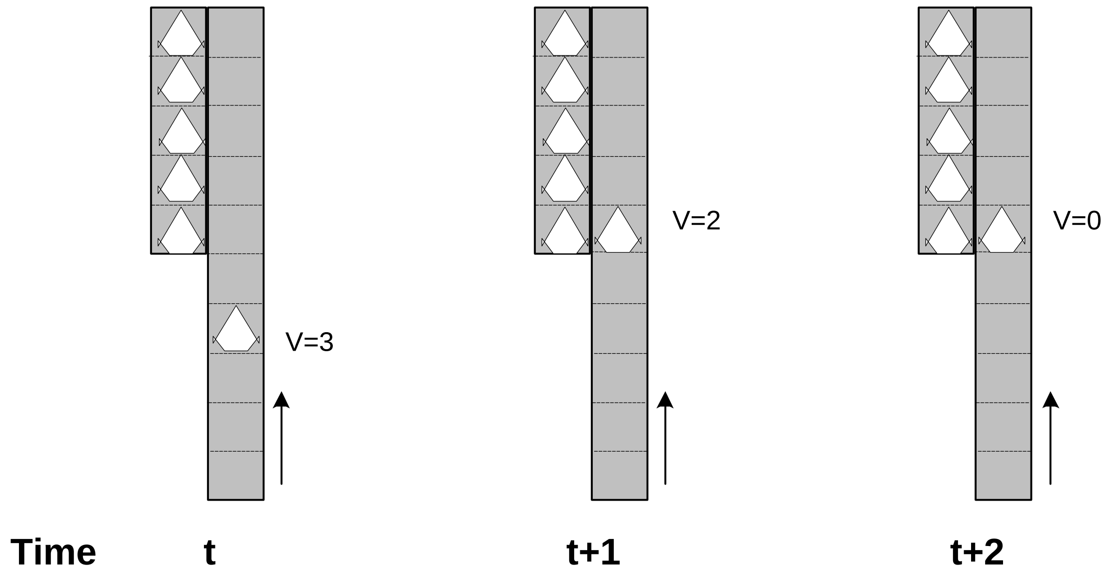

Figure 3: Turn Pocket Lanes

3.3.3.3 Look Ahead Across Links

If acceptable approach lanes are determined considering only the connectivity to the next link, some vehicles may be unable to make the required lane changes into acceptable approach lanes on short multi-lane links with multiple lane connectivity at the intersections. Looking ahead across links increases the time that a vehicle has to make a plan following lane change.

The acceptable approach lanes are determined based on a plan look ahead distance. The distance is used to determine how many links in the plan will be considered when determining the approach lanes on the current link. Plan look ahead distance is described in the next section. A distance of 262.5 meters (35 grid cells) is the default value. A value of 0.0 means that approach lanes are determined by considering the next link only.

3.4 Driver Logic

The Driver Logic parameters control how drivers and vehicles behave in traffic. Variations in behavior among drivers is accomplished by allowing certain behaviors to vary randomly within limits.

- **Deceleration Probability**. Variation in the traffic is enhanced by having each driver randomly decide whether to decelerate for no apparent reason at each time step. The probability of decelerating is a value in the range 0.0 to 1.0 [default = 0.2].

- **Lane Change Probability**. Variation in the traffic is reduced by not allowing every driver who would change lanes based on vehicle speed and gaps in the traffic to do so at each time step. This is done to prevent *lane hopping*. The probability that a driver will change lanes when speed and gaps permit is a value in the range of 0.0 to 1.0 [default = 0.99].

- **Plan Following Distance**. Plan Following Distance specifies a count of the number of cells preceding the intersection within which a vehicle will make lane changes to get in an appropriate lane to transition to the next link in its plan. Beyond this distance, lane changing decisions are based only on vehicle speed and gaps in the traffic. Within this distance, the lane required by the vehicle's plan is also taken into account. As the vehicle nears the intersection, the bias to be in the lane required to stay on plan is increased. Valid values are positive or zero [default = 70 cells].

- **Intersection Residence Time**. Intersection Residence Time specifies the number of seconds that a vehicle requires to pass through a signalized intersection. A vehicle resides in an intersection queued buffer for this amount of time and is then placed on the next link if the first cell on that link is unoccupied. It will remain in the intersection for a longer time if entry to the next link is blocked by another vehicle. Valid values are positive [default = 1 second].

- **Plan Look Ahead Distance**. The preferred lane for a vehicle to be in as it approaches an intersection depends on the connectivity from the current link to the next link in the plan. In some situations, it is advantageous for the driver to look beyond the next link to subsequent links in the plan when deciding the preferred lane. Plan Look Ahead Distance controls how far ahead the driver will look. A value of 0 indicates that the driver will not look beyond the next link. A positive value indicates that the driver will look at least one additional step beyond the next step in the plan. The number of additional links that will be considered is determined by the lengths of the subsequent links, with link lengths being summed until the accumulated distance is greater than or equal to Plan Look Ahead Distance. Valid values are positive or zero [default = 35 cells].

- **Off Plan Exit Time**. Off Plan Exit Time specifies the number of seconds a vehicle is allowed to deviate from its plan before being removed from the simulation. This prevents off-plan vehicles from wandering on the transportation network. Valid values are positive [default = 1 second].

- **Ignore Gap Probability**. Drivers at unsignalized intersections wait for a suitable gap in cross traffic before proceeding through the intersection. The deadlock that would occur when vehicles are waiting for each other at multi-way stop/yield signs is prevented by allowing each driver to ignore the gap constraint with some probability. The probability that the drivers at multi-way stop/yield signs will ignore the constraint is a value in the range of 0.0 to 1.0 [default = 0.66].

- **Gap Velocity Factor**. At unsignalized intersections and during protected movements at signalized intersections, drivers wait for a suitable gap in cross traffic before proceeding through the intersection. The number of empty cells in a suitable gap is based on the speed of the cross traffic and the gap velocity factor. The suitable gap is calculated for each lane of the cross traffic.

  Gap = Speed of Oncoming Vehicle * Gap Velocity Factor

  The gap velocity factor must be greater than 0.0. The default value is 3.0. Note that vehicles with a speed of 0 result in a suitable gap size of 0, which improves traffic flow in congested conditions.

- **Max Waiting Seconds**. Max Waiting Seconds determines the number of seconds that a vehicle will try to enter an intersection. If the vehicle has not moved from the link into or through the intersection in Max Waiting Seconds, the vehicle will abandon its plan and try an alternative movement through the intersection, if one exists. Max Waiting Seconds must be greater than 0 and should be greater than the longest red phase of the traffic controls in the simulation. The default value is 600 seconds.

3.5 Intersections

Unsignalized intersections with stop/yield traffic controls require vehicles to consider oncoming traffic before they can move onto the next link. The vehicles use the gap between the oncoming vehicles and the intersection to determine whether the intersection can be entered. If the gap is acceptable, the vehicle traverses the intersection and arrives on the destination link during a single update step in the microsimulation.

Vehicles at signalized intersections have different behavior from those at unsignalized intersections. When a vehicle enters an intersection, it is placed in a queued buffer where it resides for a specified time before exiting to the destination link. The time that vehicles spend in the queued buffer models the time necessary to traverse the intersection. Vehicles with permitted, but not protected, movements from the intersection traffic control must consider the oncoming traffic before entering the intersection.

3.5.1 Entry Conditions

In order to enter an intersection, the vehicle must do the following:

1) Be the last vehicle on the link in the current lane going toward the intersection. Only one vehicle per lane is allowed to enter the intersection in a single timestep.

2) Have a current speed ≥ the number of empty cells between the vehicle and the end of the link.

3) Satisfy the conditions of the traffic control at the intersection. The state of the traffic control indicates whether a vehicle must consider the oncoming traffic gaps.

4) Ensure that the gap between the vehicle and oncoming traffic is acceptable.

5) Ensure that the intersection buffer for the current lane is not full.

6) Ensure that the destination cell in the destination lane on the destination link is unoccupied.

A vehicle will attempt to enter an intersection if its current speed is ≥ the number of empty cells between the vehicle and the end of the link. The destination lane on the next link is determined. If possible, the destination lane is the same as the current lane. If not possible, a random lane selection is used.

The state of the traffic control (TC) at the intersection is an important factor in the decision on whether the vehicle can enter the intersection. At a signalized intersection, the TC must indicate a permitted, protected, or caution movement for the current lane in order to enter the intersection. At an unsignalized intersection, stop and yield signs impose conditions on intersection entry. The TC state may require that the distance between the intersection and the on-coming traffic (interfering lane gap) meet certain criteria before the vehicle can enter the intersection. Table 2 shows the TC states and their corresponding actions.

Table 2: Traffic Control States and Corresponding Actions

TC State	Action	Conditions
S* - Protected	Proceed	None
S - Wait	Stop	None
S - Permitted	Evaluate	G_i on IL (Interfering Lanes)
S - Caution	Proceed	None
U** -None	Proceed	None
U - Stop	Wait Evaluate	Stopped < 1 Timestep G_i on IL, Stopped ≥ 1 Timestep
U - Yield	Evaluate	G_i on IL

* S = Signalized intersection

** U = Unsignalized intersection

The interfering lane gap (G_i) is the distance between the oncoming vehicle and the intersection. The oncoming vehicle must be on a link connected to the intersection, which limits the look-back distance for oncoming traffic to the length of a single link. The speed of the oncoming vehicle (V_{OV}) and the Gap Velocity Factor (GVF) are used to calculate the Desired Gap.

Desired Gap (G_d) = V_{OV}*GVF

On links where the desired gap is greater than the number of cells on the link, the number of cells on the link is used as the desired gap.

$G_i \geq G_d$, Interfering Gap Acceptable

$G_i < G_d$, Interfering Gap Not Acceptable

Note that for an oncoming vehicle with speed of 0, G_d will be 0, which allows movement through intersections in congested conditions where both G_d and $G_i = 0$.

The vehicle can enter the intersection only when the interfering gaps are acceptable ($G_i \geq G_d$) (Figure 4).

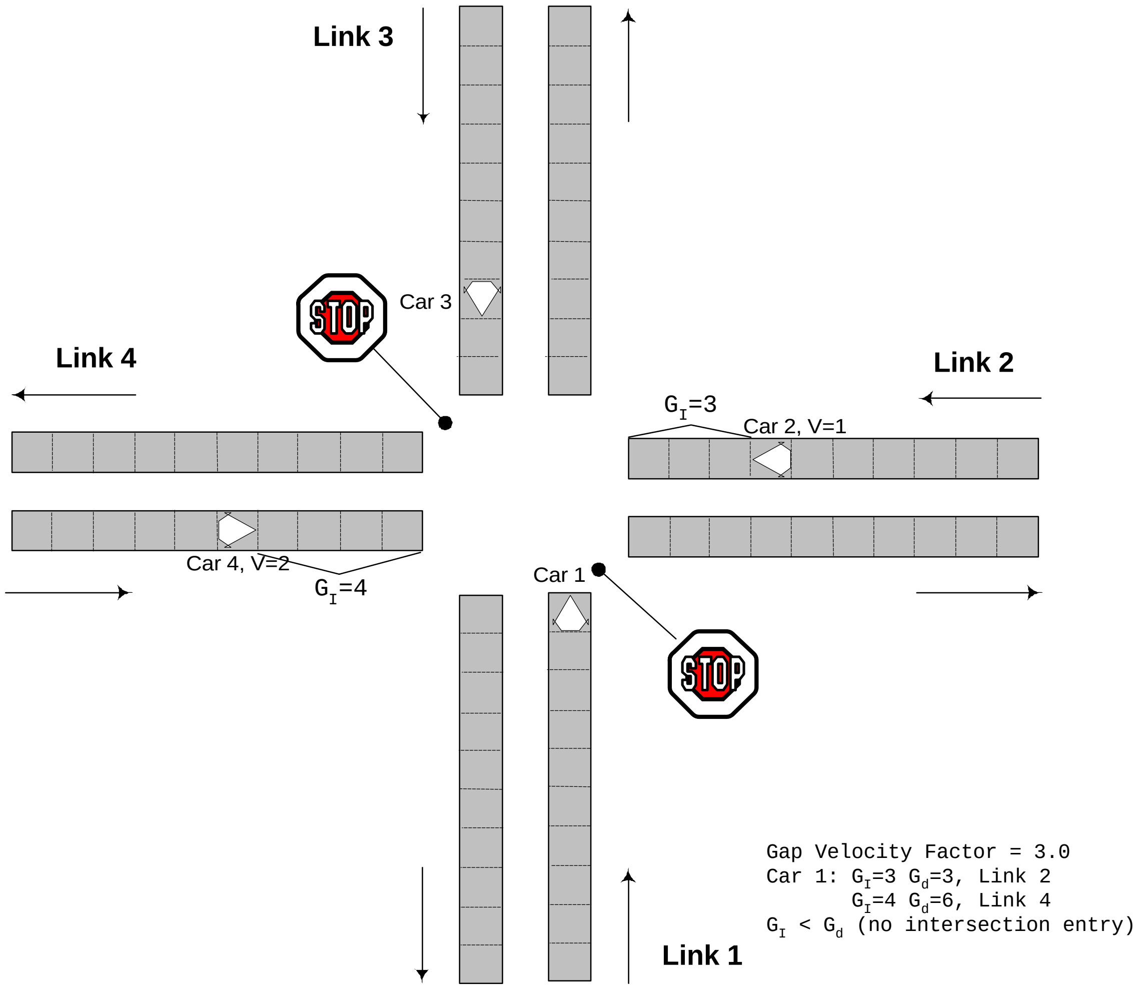

Figure 4: Intersection Entry -- Interfering Lane Gap

If the traffic control for the intersection is signalized, the vehicle does not traverse the intersection in the current microsimulation timestep. Signalized intersections maintain internal queued buffers where vehicles are placed to traverse the intersection. Each intersection has one queued buffer for each incoming lane. If the conditions of the signalized traffic control have been satisfied, a vehicle must check whether the appropriate buffer has space to receive the vehicle. If so, the vehicle is removed from the incoming link and placed in the intersection buffer for a specified wait period (currently one timestep). After the time period has expired, the vehicle will exit from the buffer to the first cell on the destination link if the cell is vacant. If not, the vehicle waits in the intersection buffer until the cell becomes vacant. The buffers have a fixed size so that if the buffer is full, the vehicle cannot enter the intersection and must wait on the link.

At unsignalized intersections, the vehicles are capable of entering and exiting the intersection in a single timestep. Therefore, if the conditions of the unsignalized traffic control have been satisfied for intersection entry, a vacant cell on the destination link in the destination lane must be available for the vehicle to enter the intersection. The vehicle's current speed is used to determine which cell to reserve on the destination link. If the primary destination cell is unavailable, the next cell closer to the intersection is tried. The process continues until an available cell is found, or until all of the cells between the intersection and the primary destination cell are tried. A marker is placed in the destination cell to reserve the cell.

After all of the vehicles have determined if they can enter the intersections and movement for the other vehicles on the grids is executed, vehicles are transferred from links to intersection queued buffers (signalized intersections) or from links to other links (unsignalized intersections).

3.5.2 Signalized Intersection Exit

Vehicles exit from the intersection queued buffers when their residence time in the buffer is greater than the intersection residence time described in Section 3.4. Vehicles exit from the queued buffer onto the first cell in the destination lane on the destination link. Exiting vehicles reserve their destination cell before vehicles on links calculate movement, which gives the vehicles exiting from intersection buffers precedence over vehicles on the links. Vehicles are transferred from the buffers to their reserved destination cells after movement changes for all the vehicles are executed. The speed of the vehicle does not change during intersection entry/exit at a signalized intersection. Vehicles are placed in the first cell on the destination link with the same velocity that they entered the intersection buffer.

4. PLANS

4.1 Preprocessing

The microsimulation creates vehicles from a set of travel plans produced by the TRANSIMS Route Planner. Each plan begins and ends at a parking place. The TRANSIMS Route Planner produces plans on the regional transportation network. The microsimulation for Case Study 1 uses a study area that is a subset of the regional network. The plan set produced by the Planner is preprocessed to limit the plan set to plans that enter the microsimulation study area during the simulation time interval, to truncate the plans to the series of links in the microsimulation study area, and to sort the plans by time of entry into the microsimulation study area.

4.2 Microsimulation Use

The microsimulation reads these pre-processed, time-sorted files and creates vehicles with associated plans. The microsimulation uses the preprocessed and sorted plan set to obtain the starting link of each plan. The vehicle is then placed in a parking place queue on the starting link. As simulation time advances, the microsimulation removes each vehicle from the parking place queue at the appropriate starting time, and places the vehicle on the link. The vehicle then traverses the roadway network following the links in its plan until it reaches a destination parking place in the study area or it completes the sequence of links in its plan that are in the microsimulation study area.

All vehicles are not created at simulation initialization; instead, vehicles are created periodically at user-specified intervals of time. Only those plans with a departure time during the next time interval are read and instantiated. This just-in-time scheme for creating vehicles and plans conserves computer memory because only those vehicles currently on the roadway, or about to enter it from a parking place, exist simultaneously in the simulation.

It is desirable for the simulation to reach normal traffic flow conditions as rapidly as possible. This is facilitated by placing vehicles on the roadway when the simulation is initialized according to where the plans predict they will be at the simulation starting time. Vehicles with a planned departure time later than the simulation starting time are placed in the queue at the plan origin parking place. They enter the roadway at the appropriate time as described in Section 5.1. Vehicles with a plan that predicts they will arrive at their destination before the simulation begins are never created. Vehicles with a plan that indicates they will be enroute at the simulation starting time are placed on the predicted link. The cell position is interpolated according the traversal time for the link, and the lane is chosen randomly. If the indicated cell is already occupied with a vehicle, the grid is searched upstream for an available cell. If all cells upstream are occupied, the grid is searched downstream for an unoccupied cell. If all cells on the link are occupied, a warning message is printed and the vehicle is deleted. No attempt is made to find an available cell on an adjacent link. If the vehicle is on the last step of its plan and its interpolated position is beyond its destination parking place, the vehicle is not created. Distributed links require extra care to ensure that the vehicle is created only on the CPU containing the portion of the grid indicated by the interpolated vehicle position.

4.3 Plan Following Problems

4.3.1 Off-plan Vehicles

A vehicle that is not in an acceptable approach lane when it is ready to enter an intersection cannot follow its assigned plan; it is marked as an *off-plan vehicle*. The timestep when the vehicle will try to exit from the simulation is calculated using the off-plan exit time described in Section 3.4, and a new destination link is chosen from the links that are connected to the vehicle's current lane.

New destination links are randomly chosen for off-plan vehicles until the current timestep is equal to the calculated exit timestep. Then, the vehicles are removed from the simulation at the nearest parking place.

4.3.2 Abandon Plan

Vehicles that are trying to enter an intersection and have not moved for a specified period of time abandon their plans and, if possible, choose a different destination link. Max Waiting Seconds defines the time period (Section 3.4). These vehicles are marked as off-plan and are removed at the nearest parking place. Allowing vehicles to become off-plan after a specified waiting period is necessary to prevent traffic gridlock in congestion.

5. VEHICLE ENTRY/EXIT (PARKING)

Parking places are created when the transportation network is constructed prior to the beginning of the simulation. A database table specifies the locations and characteristics of the parking places in the study area. Parking places include a queue in which vehicles wait until their plan indicates they should enter the simulation. Methods for exiting and entering the parking place are provided.

5.1 Exit from Parking Places

Vehicles are removed from a queue in the parking place and placed on the roadway. The parking queue is checked to find a vehicle whose departure time has come. The appropriate grid for the planned direction of travel is determined, and the grid is searched upstream for a distance of V_{Max} cells. If a vehicle is found in a lane, that lane and adjacent lanes are eliminated from consideration. All lanes are searched and if a lane is available, the vehicle is removed from the queue and placed on the lane at the cell corresponding to the parking place location.

5.2 Entering Parking Places

Vehicles are removed from the roadway at destination parking places by checking all of the cells in all lanes downstream from a parking place for a distance of $V_{GlobalMax}$ cells. If a vehicle is found on the last step of the current leg of its plan and with this parking place as its destination, the vehicle is removed from the roadway. Currently, the vehicle is deleted from the simulation; however, in future IOCs, when multi-leg plans are supported, the vehicle will enter the parking place to wait for the departure time for the next leg of the plan.

5.3 Boundary Parking

Parking places for links in the microsimulation buffer region are not specified in the database tables. Instead, a single generic parking place is created at the mid-point of each of the links in the buffer area. These boundary parking places are the locations where vehicles originating outside the study area enter the simulation, and where vehicles with destinations outside the study area exit the simulation.

6. PARALLEL COMPUTATION

The TRANSIMS microsimulation runs on multiple CPUs to maximize the computational speed. Updating of vehicle positions can then be done in parallel on the individual CPUs. This method is faster than a single, sequential update algorithm on transportation networks with a large number of vehicles.

6.1 Transportation Network Partition

The transportation network is partitioned among the CPUs, with each CPU receiving a set of nodes and links (Figure 5). An orthogonal bisection algorithm is used to partition the network nodes among the compute nodes. This algorithm uses the x and y coordinates of the nodes and a cost function for each node based on the number of cells on the links attached to the node.

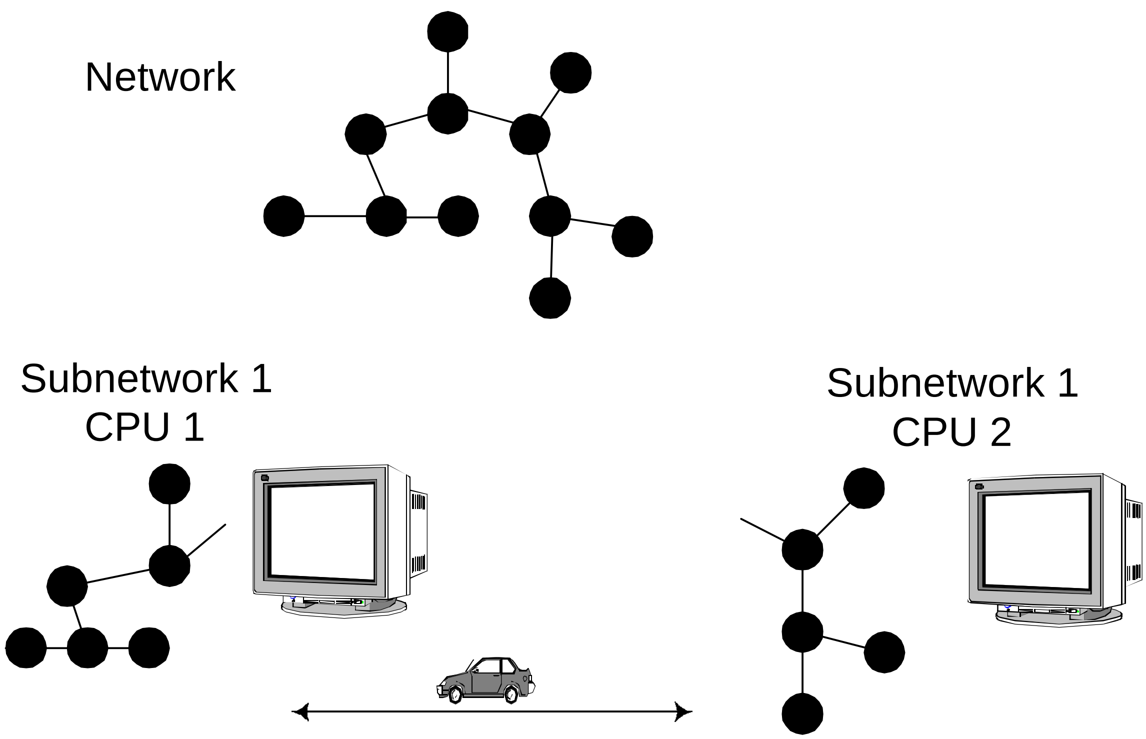

Figure 5: Transportation Network Partition

6.2 Distributed Links and Boundary Information Flow

Links that connect nodes that reside on different CPUs are split in the middle (Figure 6). These links are distributed links. Each CPU is responsible for one-half of the link. Each distributed link is assigned the number of active grid cells belonging to the given CPU. This is necessary to accurately divide links with an odd number of cells. The area in the middle of the distributed links is called a boundary area. The width of the boundary area is currently $V_{GlobalMax}$ (5) cells. The maximum distance (forward or backward on a link) that can be used for gap calculations is limited to the boundary width on distributed links.

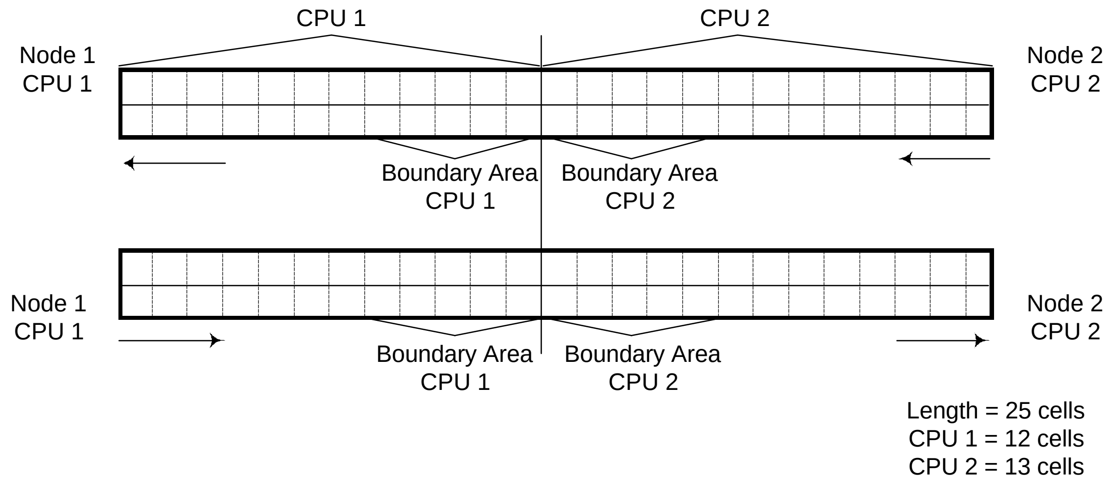

Figure 6: Distributed (Split) Link

Vehicles are transferred between the CPUs as they traverse these split links. Each split link introduces a message-passing delay during the update sequence because messages must be passed between the CPUs for vehicles that are crossing the split links. Two types of messages must be exchanged between CPUs with distributed links.

1) Vehicle Migration Messages – Messages for vehicles transferred to the other part of the link on a different CPU.

2) Boundary Exchange Messages – Messages containing information about vehicle positions in the boundary area of a link.

Vehicle migration messages occur for all vehicles that have completed traversal of a CPU's active cells. The information about the vehicle is put into a message and sent to the CPU that owns the other half of the distributed link, after which the vehicle is removed from the originating CPU. Upon receipt of the message, the other CPU creates a vehicle using the information in the message and places it at the appropriate position on its half of the distributed link.

Exchange of boundary information between CPUs is called a boundary exchange. Boundary exchange messages are necessary to correctly calculate position changes (movement and lane changes) for vehicles in a CPU's boundary area. Information about vehicles in the next five cells (or preceding five cells, depending on the direction of traffic flow) is necessary to execute the appropriate gap calculations for lane changes and movement. Each CPU maintains a list of its distributed links and of the CPU owners of the other half of the links. Boundary exchanges must be done before lane changes and again before vehicle movement. The exchanges are initiated by each CPU at the appropriate time. Each CPU waits until it receives all of the boundary exchange messages from neighboring CPUs.

6.3 Parallel Computation Sequence and Synchronization Points

The TRANSIMS microsimulation is a distributed object simulation using a master/slave(s) paradigm. The master process starts the slave processes, handles the initialization sequence, and then serves as a synchronization point for the slave processes. The slave processes do all of the work in the simulation. After initialization, each slave process completes successive update cycles until the end of the specified simulation run. In each slave update cycle,

- new vehicles are created from plans at the appropriate time and placed on the transportation network,
- traffic controls are updated,
- new positions are calculated for existing vehicles, and
- vehicles reaching their destination or exiting from the microsimulation study area are removed.

One update cycle is called a timestep. All output information from the microsimulation is generated in parallel by the slave(s). The slave processes synchronize with the master process at the beginning of each timestep.

6.3.1 PVM and CPU Usage

The microsimulation uses a parallel execution environment called Parallel Virtual Machine (PVM). PVM is widely available with no license fee. PVM provides a communication mechanism between processes. These processes can be on the same CPU/host or on different CPUs/hosts. Each microsimulation master/slave process becomes a PVM task. Messages between master and slave(s) and among slaves utilize PVM as the underlying message handler.

The microsimulation can be run using two PVM configurations:

1) Multiprocessor
2) Workstation/LAN

The multiprocessor configuration is used to run the microsimulation processes on a machine with multiple CPUs. Communication between the processes will utilize shared memory.

The workstation/LAN configuration is used to run the microsimulation on single-CPU workstations on a local area network (LAN). There must be at least two hosts in the PVM configuration. One of the hosts will be used for the master process. All of the other hosts will be used for slave processes, with one slave process per host. The LAN is used for communication between the microsimulation processes.

6.3.2 Initialization Sequence

The master process starts first, enrolls itself as a PVM task, reads network information from the database, and constructs a copy of the transportation network. The master is then ready to create and initialize the slave processes using the following sequence:

1) Start slave processes as PVM tasks.

2) Partition the transportation network over the slaves.

3) Send each slave its list of nodes and links.

4) Send each node on each slave a list of links that are connected to the node.

5) Tell each slave to construct its transportation subnetwork from information in the database.

6) Send distributed edge information to slaves.

7) Tell slaves to create grids on the transportation network, to queue initial vehicles on parking places, and to do initial placement of vehicles on the links at the given simulation start time.

8) Tell slaves to create intersection queued buffers for signalized intersections.

6.3.3 Computation Sequence and Synchronization Points

After the initialization sequence is complete, the master starts the simulation by telling the slaves to execute the first timestep. The master process waits until all of the slaves complete execution of a timestep then sends a message to the slaves to execute the next timestep. The master sends a message to the slaves to shut down when the requested number of timesteps has been executed.

The slaves execute the following sequence during a timestep:

1) Increment the timestep value.

2) Read the next group of plans, if necessary.

3) Update the traffic signal states.

4) Reserve the destination cells for vehicles ready to exit intersection buffers.

5) Do lane changes for vehicles on grids.

6) Place vehicles ready to exit from parking place queues on the appropriate links.

7) Do boundary exchange with neighboring CPUs.

8) Determine which vehicles will enter intersections. Reserve a destination cell for vehicles entering and exiting intersections during this timestep (unsignalized intersections). Check the intersection buffer status (okay/full) for those vehicles entering signalized intersections.

9) Mark as off-plan, those vehicles that are trying to enter intersections on lanes that have no connectivity to the next link in their plans.

10) Move vehicles on the links. Calculate new speeds for all vehicles, and move them to new positions on the grid. Reset speed constraints that were set in the lane change method. For vehicles entering intersections: a) mark for transfer to the intersection, b) move to the last cell on the current link, and c) reset the speed to the number of cells moved on the current link plus the number of cells to the destination cell on the new link.

11) Mark vehicles to be transferred to another CPU on distributed links.

12) Remove vehicles entering parking places from the links.

13) Transfer vehicles to other CPUs from distributed links.

14) Receive transferred vehicles from other CPUs on distributed links.

15) Send boundary information to neighboring CPUs.

16) Remove vehicles from intersection buffers and place them in reserved cells on links.

17) Transfer vehicles entering intersections to queued intersection buffers (signalized intersections) or to reserved cells on destination links (unsignalized intersections).

18) Delete vehicles on distributed links that have been transferred to other CPUs.

19) Remove unused cell markers that were used to reserve cells on destination links.

20) Read and process boundary information from neighboring CPUs.

21) Synchronize with the master and other slaves by sending a timestep complete message to master.

7. SIMULATION OUTPUT

Three general types of microsimulation output are available:

1) Snapshot data
2) Summary data
3) Event data

The exact nature and frequency of output collection is specified by the user prior to beginning the simulation through the creation of output specification tables. Output collection is initialized in each slave process when the process is started.

Snapshot data provides detailed information regarding how the microsimulation evolves in time. On links, information about the position, velocity, and status of each vehicle on the link is recorded. At nodes, the signal state and status of intersection buffers is recorded. Snapshot data may be recorded at each timestep, or less frequently.

Summary data provides aggregated information about the simulation. Spatial summary data records vehicle counts and mean velocities in *boxes* that are several cells wide. Temporal summary data records means and variances of link traversal times by vehicles. Summary data is sampled periodically and recorded less frequently, as specified by the user.

Event data is not recorded periodically, but it is recorded each time an event occurs. The events are triggered by a state change that occurs in a vehicle. The six vehicle state change events are:

1) Off Plan – the vehicle is unable to follow its plan. This event may occur when a vehicle is unable to execute lane changes to make the appropriate turns at intersections in order to follow its plan. The event may also occur when a vehicle decides to abandon its plan because link congestion has prevented movement for long periods of time.

2) At Dead End – The vehicle is on a link with no connectivity to any other links.

3) Entry Into Study Area – The vehicle has entered the study area. A vehicle can enter the study area from a parking lot on a study area link. A vehicle can also enter the study area by traveling from a buffer area link to a study area link.

4) In Study Area – The vehicle is in the study area. This status remains set as long as a vehicle is traveling on a study area link.

5) Exit From Study Area – The vehicle has exited the study area. A vehicle can exit the study area by entering a parking lot on a study area link or by traveling from a study area link to a buffer area link.

6) Error In Plan – The vehicle has an invalid plan. This event occurs when an error in the plan is detected, such as a link sequence where there is no connectivity between the links in the transportation network.

A vehicle can enter the microsimulation in three ways:

1) The vehicle is on a study area link at the simulation start time. (Entry & InStudyArea, Status value = 20)

2) The vehicle enters the study area by leaving a parking location. (Entry & InStudyArea, Status value = 20)

3) The vehicle crosses the boundary into the study area from the outside. (Entry & InStudyArea, Status value = 20)

A vehicle can leave the microsimulation in four ways:

1) The vehicle is in the study area when the simulation ends. (Exit & InStudyArea, Status value = 24)

2) The vehicle leaves the study area by entering a parking location. (Exit & InStudyArea, Status value = 24)

3) The vehicle crosses the boundary out of the study area to the outside. (Exit & InStudyArea, Status value = 24)

4) The vehicle reaches a dead end within the study area. (Exit & InStudyArea & AtDeadEnd, Status value = 26)

The combination (Entry & Exit & InStudyArea, Status value = 28) only occurs for vehicles that enter the study area just as the simulation ends. The OffPlan and BadPlan status values can be combined with any other status values except each other (even if the vehicle is not in the study area). An OffPlan status value adds 1 to the previously listed status values. A BadPlan status value adds 32 to the previously listed status values. Similarly, AtDeadEnd may occur in combination with any other status, but never by itself. AtDeadEnd adds 2 to the other status values. Table 3 summarizes the possible vehicle status values:

Table 3: Possible Vehicle Status Values

Bad Plan	InStudy Area	Exit	Entry	AtDeadEnd	OffPlan	Vehicle Status
					X	1
				X	X	3
	X					16
	X				X	17
	X			X		18
	X			X	X	19
	X		X			20
	X		X		X	21
	X		X	X		22
	X		X	X	X	23
	X	X				24
	X	X			X	25
	X	X		X		26
	X	X		X	X	27
	X	X	X			28
	X	X	X		X	29
	X	X	X	X		30
	X	X	X	X	X	31
X						32
X				X		34
X	X					48
X	X			X		50
X	X		X			52
X	X		X	X		54
X	X	X				56
X	X	X		X		58
X	X	X	X			60
X	X	X	X	X		62

8. REFERENCES

[1] Nagel, K. and Schreckenbereg, M. (1992), “A cellular automaton model for freeway traffic,” J. Phys. I France, Vol. 2, pg. 2221.

[2] Nagel, K. (1996), “Particle hopping models and traffic flow theory,” Phys. Rev. E, Vol. 53(5), pg. 4655.

[3] Rickert, M., Nagel, K., Schreckenberg, M. and Latour, A. (1996), “Two lane traffic simulations using cellular automata,” Physica A, Vol. 231, pg. 534.

[4] Wagner, P., Nagel, K. and Wolf, D.E. (1997), “Realistic multi-lane traffic rules for cellular automata,” Physica A, Vol. 234, pg. 687.

[5] Nagel, K., Pieck, M., Leckey, S., Stretz, P., Donnelly, R. and Barrett, C.L., “TRANSIMS traffic flow characteristics,” TRB Paper 981332, 1998.